

## BAB II

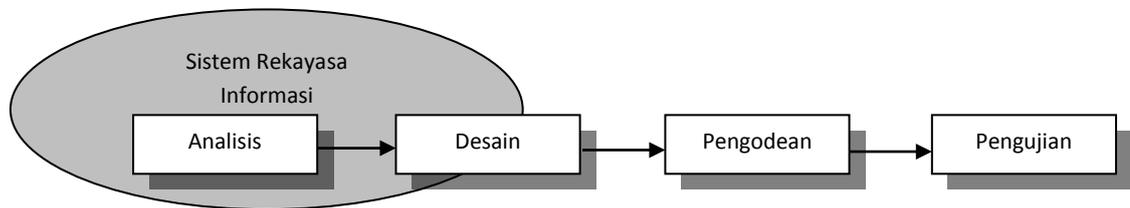
### LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Didalam penelitian ini, penulis akan menguraikan tentang apa yang akan penulis teliti agar dapat memudahkan dan menjelaskan secara lebih rinci tentang apa saja variabel yang telah penulis teliti. Berdasarkan referensi dari literatur-literatur yang berisi teori-teori yang berkaitan dengan masalah yang akan dibahas mengenai Sistem Informasi *E-Filing* Dokumen Permintaan Posting Pada *Visa Master Card Operation* Bank Mandiri. Penulis melakukan studi kepustakaan melalui sumber buku, jurnal-jurnal terkait, maupun pada sumber lainnya sebagai dasar pembahasan lebih lanjut.

##### 2.1.1. Konsep Model Sistem Informasi

Model pengembangan sistem yang digunakan dalam Skripsi ini adalah model *Waterfall*. Menurut Sukamto dan Shalahuddin (2013a:31) “Model *waterfall* adalah model SDLC yang paling sederhana. Model ini hanya cocok untuk pengembangan perangkat lunak dengan spesifikasi yang tidak berubah-ubah”.



**Gambar II.1**

#### **Ilustrasi Model *Waterfall***

Sumber : Sukamto dan Shalahuddin (2013c:28)

Berikut ini adalah tahapan-tahapan dari model *waterfall* :

#### 1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

#### 2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

#### 3. Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

#### 4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

#### 5. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

Dengan berbagai kelemahan yang dimiliki model *waterfall* tapi model ini telah menjadi dasar dari model-model yang lain dalam melakukan perbaikan model pengembangan perangkat lunak. Hal positif dari model ini adalah struktur tahap pengembangan sistem jelas, dokumentasi dihasilkan di setiap tahap pengembangan, dan sebuah tahap dijalankan setelah tahap sebelumnya selesai dijalankan.

Sistem secara umum didefinisikan sebagai kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu sebagai satu kesatuan. (Agus Mulyanto, 2009 : 1)

Mendefinisikan sistem dalam bidang sistem informasi sebagai “sekelompok komponen yang saling berhubungan, bekerja sama, untuk mencapai tujuan bersama dengan menerima proses *input* serta menghasilkan *input* dalam proses transformasi yang teratur”.

Dengan demikian pengertian sistem dapat disimpulkan sebagai suatu prosedur atau elemen yang saling berhubungan satu sama lain dimana dalam

sebuah sistem terdapat suatu masukan, proses dan keluaran, untuk mencapai tujuan yang diharapkan. (Agus Mulyanto, 2009 : 2)

Suatu sistem mempunyai beberapa karakteristik, yaitu komponen atau elemen (*component*), batas sistem (*boundary*), lingkungan luar sistem (*environment*), penghubung (*interface*), masukan (*input*), pengolah (*process*), keluaran (*output*), sasaran (*objective*), atau tujuan (*goal*). (Agus Mulyanto, 2009 : 3).

Karakteristik sistem mempunyai beberapa komponen diantaranya yaitu :

1. Komponen Sistem (*Components*)

Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem yang saling berinteraksi, artinya saling bekerja sama membentuk satu kesatuan.

2. Batas Sistem (*Boundary*)

Merupakan daerah yang membatasi suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan.

3. Lingkungan Luar Sistem (*Environments*)

Lingkungan luar dari suatu sistem adalah apapun diluar batas sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat merugikan atau menguntungkan sistem tersebut.

4. Penghubung (*Interface*)

Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem ke subsistem yang lainnya. Dengan penghubung, satu

subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

#### 5. Masukan (*Input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

#### 6. Keluaran (*Output*)

Merupakan hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem.

#### 7. Pengolahan (*Process*)

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan mengubah masukan menjadi keluaran.

#### 8. Sasaran (*Objectives*) dan Tujuan (*Goal*)

Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya.

Ada beberapa bentuk klasifikasi system dilihat dari sudut pandang, diantaranya adalah (Kristanto, 2008 : 5):

#### 1. Sistem Abstrak (*Abstrack System*) dan Sistem Fisik (*Physical System*)

Sistem abstrak adalah sistem yang tidak bisa dilihat secara mata dan biasanya system ini berupa pemikiran atau ide-ide. Sistem fisik merupakan system yang biasa dilihat secara mata bisa dan biasanya digunakan oleh manusia.

2. Sistem Alamiah (*Natural System*) dan Sistem Buatan Manusia (*Human Made System*)

Sistem alamiah adalah sistem sistem yang terjadi melalui proses alam, tidak dibuat manusia.

3. Sistem Tertentu (*Deterministic System*) dan Sistem Tak Tentu (*Probabilistic System*)

Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan.

4. Sistem Tertutup (*Closed System*) dan Sistem Terbuka (*Open System*)

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya, oleh karena itu perlu adanya sistem pengendalian yang dapat menjaga agar pengaruh tersebut hanya berupa pengaruh yang baik saja.

#### **A. PHP (*Hypertext Preprocessor*)**

Menurut Sidik (2012:4) PHP merupakan secara umum dikenal sebagai bahasa pemrograman *script-script* yang membuat dokumen HTML secara *on the fly* yang dieksekusi di server *web*, dokumen HTML yang dihasilkan dari suatu aplikasi buatan dokumen HTML yang dibuat menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemrograman *server side*.

Menurut Adelheid dan Nst (2012:2) “PHP merupakan bahasa berbentuk *script* yang ditempatkan dalam *server* dan dieksekusi di dalam *server* untuk selanjutnya ditransfer atau dibaca oleh *client*.”

Dari pendapat diatas maka dapat disimpulkan bahwa PHP adalah bahasa program berbentuk skrip dan jenis bahasa pemrograman yang *open source*.

## **B. MYSQL**

Menurut Nugroho (2011:29) MySQL merupakan *database* yang berbasis *server*. Anda dapat menggunakan *database* MySQL apabila memiliki hak akses didalamnya. Hal ini seperti halnya pada saat anda hendak menggunakan klien MySQL untuk masuk pada *server* MySQL, dengan menggunakan program PHP, anda dapat mengakses *database* MySQL dengan terlebih dahulu menggunakan koneksi.

Menurut Sidik (2012:333) “ MySQL merupakan *software database* yang termasuk paling populer dilingkungan linux, kepopuleran karena ditunjang karena performansi *query* dari databasenya yang saat itu bisa dikatakan paling cepat dan jarang bermasalah.”

Dari pendapat diatas maka dapat disimpulkan bahwa MySQL adalah *database* yang berbasis *server* yang memudahkan penyimpanan, perubahan dan akses informasi.

## **C. Xampp**

Menurut Sidik (2012:72) “ XAMPP (X(Windows/Linux) Apache MySQL PHP dan Perl) merupakan paket *server web* PHP dan *database* MySQL yang paling

populer dikalangan pengembangan *web* dengan menggunakan PHP dan MySQL sebagai *databasenya*.”

Menurut Madcoms (2011:5) “ XAMPP dan AppServ adalah salah satu paket *software web server* yang terdiri dari *Apache, Mysql, Php dan PhpMyAdmin*.”

Dari pendapat diatas maka dapat disimpulkan bahwa XAMPP adalah aplikasi *webserver* yang menggabungkan PHP, *Webserver Apache* dan *database MySQL*.

#### **D. Hosting**

Menurut Madcoms (2011:6) “ Merupakan tempat dalam *server* komputer yang digunakan untuk menempatkan data dan *file web*, sehingga data dan *file web* tersebut dapat diakses oleh semua *user* pengguna *internet*.”

Menurut Herudi (2011) “ *Hosting* adalah penyewaan tempat untuk menampung data-data yang diperlukan oleh sebuah blog atau *website* yang memiliki *top-level* domain sendiri sehingga dapat diakses lewat *internet*.”

Dari pendapat diatas maka dapat disimpulkan bahwa *hosting* adalah tempat didalam *server* komputer yang berguna untuk menampung atau menempatkan data-data sehingga dapat diakses oleh semua *user*.

#### **E. Domain**

Menurut Madcoms (2011:7) “ *Domain* adalah nama unik atau alamat yang digunakan untuk mengidentifikasi nama *server* komputer, seperti *web server* atau *email server* di *internet*.”

Menurut Herudi (2011) “ *Domain* berfungsi untuk mempermudah pengguna *internet* pada saat melakukan akses ke *server*.”

Dari pendapat diatas maka dapat disimpulkan bahwa domain adalah sebuah nama atau alamat untuk mempermudah mengakses ke server.

### 2.1.2. UML (Unified Modelling Language)

Menurut Rosa dan Shalahuddin (2011:113) “ UML (*Unified Modelling Language*)” adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analis dan desain serta menggambarkan arsitektur dalam pemrograman berorientasi objek.”

Menurut Widodo dan Herlawati (2011:6) “*Unified Modelling Language* (UML) adalah bahasa pemodelan standar.” Menurut Chonoles dalam Widodo dan Herlawati (2011:6) mengatakan sebagai bahasa , berarti *UML* memiliki sintaks dan semantik.

UML diaplikasikan untuk maksud tertentu menurut Widodo dan Herlawati (2011:6) biasanya antara lain untuk:

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

Adapun diagram-diagram yang ada didalam UML adalah sebagai berikut:

#### a. *Use Case Diagram*

Menurut Widodo dan Herlawati (2011:21) Mengartikan use case sebagai urutan langkah-langkah yang secara tindakan saling terkait (skenario), baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal. *Use case* digambarkan dalam bentuk ellips/oval. Use case sangat menentukan karakteristik sistem yang kita buat.

Menurut Rosa dan Shalahuddin (2011:13) “ Yang dimaksud dengan use case adalah pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat.”

*b. Activity Diagram*

Menurut Widodo dan Herlawati (2011:11) “ *Activity Diagram* adalah tipe khusus dari *diagram* status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem.”

Menurut Rosa dan Shalahuddin (2011:134) “ Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis.

*c. Sequence Diagram*

Menurut Widodo dan Herlawati (2011:11) Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.

Menurut Rosa dan Shalahuddin (2011:137) “ Yang dimaksud dengan *sequence diagram* adalah suatu *diagram* yang menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.”

d. *Class Diagram*

Menurut Widodo dan Herlawati (2011:11) *Diagram* kelas bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada permodelan sistem berorientasi object. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif.

Menurut Rosa dan Shalahuddin (2011:122) “ Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.”

e. *Deployment Diagram*

Menurut Widodo dan Herlawati (2011:12) Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*). Memuat simpul-simpul beserta komponen-komponen yang ada di dalamnya. Diagram *deployment* berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*).

Menurut Rosa dan Shalahuddin (2011:149) “ Diagram *deployment* atau *deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi.”

### 2.1.3. ERD (*Entity Relationship Diagram*)

Menurut Fathansyah (2007:79) memberikan batasan bahwa *Diagram Entity-Relationship* (Diagram E-R) adalah “*Model Entity Relationship* yang berisi komponen-komponen Himpunan Entitas dan Himpunan Relasi yang masing-

masing dilengkapi dengan atribut-atribut yang merepresentasikan seluruh fakta dari ‘dunia nyata’ yang kita tinjau”.

#### 2.1.4. LRS (*Logical Record Structure*)

*Logical Record Structure* (LRS) merupakan representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil relasi antar himpunan entitas. *File record* pada LRS ditempatkan dalam kotak. LRS terdiri dari *link-link* diantara tipe *record* lainnya, banyaknya *link* dari LRS yang diberi nama oleh *field-field* yang kelihatan pada kedua link tipe *record*.

Komponen yang membentuk *Model Entity Relationship* adalah :

1. Entitas (*Entity*), merupakan individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain. Simbol dari entitas adalah persegi panjang.
2. Atribut (*Attributes/properties*), menguraikan entitas dimana mereka dihubungkan. Dan setiap entitas pasti memiliki atribut yang mendeskripsikan karakteristik (*properties*) dari entitas tersebut. Simbol dari atribut adalah *ellips*.
3. Relasi (*Relationship*), menunjukkan adanya hubungan diantara sejumlah entitas yang berasal dari himpunan yang berbeda.
4. *Indicator type*, terdiri dari dua macam yaitu :
  - a. *Associative entities* (juga dikenal sebagai *intersection entities*), adalah entitas yang digunakan oleh rekanan dua entitas atau lebih untuk menyatukan suatu hubungan banyak-ke-banyak.

- b. *Subtypes entities*, digunakan didalam hierarki generalisasi untuk menyajikan suatu *subset* kejadian dari entitas orang tua, yang disebut *supertype*, tetapi yang memiliki atribut atau hubungan yang berlaku hanya untuk *subset*.
5. *Cardinality Ratio*, menentukan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain.

Macam dari *Cardinality Ratio* adalah :

- a. *One to one* (1:1), yang berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B.
- b. *One to many* (1:M), yang berarti setiap entitas pada himpunan entitas A berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya.
- c. *Many to one* (M:1), yang berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, tetapi tidak sebaliknya.
- d. *Many to many* (M:M), yang berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan demikian juga sebaliknya.

## 2.2 Penelitian Terkait

Menurut Rizki Alflasca pascapraharastyan, Antok Supriyanto, Pantjawati Sudarmaningtyas (2014:139) bahwa:

Pada Rumah Sakit Bedah Surabaya (RSBS) kegiatan pengarsipan masih dilakukan secara semi komputerisasi dalam hal ini peranan komputer hanya sebatas pada pembuatan surat, sehingga sering sekali dokumen tidak tertata rapi, rusak atau hilang, misalnya pada saat dokumen dibutuhkan untuk ditinjau

kembali, karyawan sulit menemukan dokumen tersebut, dan mengakibatkan pendistribusian surat atau dokumen kepada yang bersangkutan tidak tepat waktu, Masalah lain yang terdapat pada RSBS yaitu belum terdapat penentuan arsip atau dokumen yang telas memaski masa kadaluwarsa. Selain itu RSBS juga membutuhkan sistem kearsipan yang aman serta mudah diakses untuk mendukung kegiatan pengarsipan di RSBS tersebut.

Menurut Iwan S, Yoppy Mirza Maulana, Tutut Wuriyanto (2014:158)

bahwa:

Pada Perusahaan Daerah Pasar Surya (PDPS) permasalahan terletak pada proses penyimpanan (rekaman/pencatatan dan pengarsipan), Selama ini proses penyimpanan dilakukan baik dan darpi dengan cara mencatat pada buku agenda maupun buku ekspedisi dan mengarsip surat berupa *hardcopy* salinan/tembusan surat kebeberapa folder penyimpanan. Namun untuk dapat menghasilkan laporan dari rekaman buku agenda maupun buku ekspedisi masih kurang efisien. Petugas harus mengecek, memilah, dan mengelompokan data rekaman satu per-satu sesuai jenis laporan yang akan disajikan.